快速排序

与算法导论(第三版)章节的对应关系

- Chapter 7. 快速排序
- 涉及知识点包括：
  - 快速排序的描述
  - 快速排序的性能
  - 快速排序的随机化版本
  - 快速排序的期望运行时间

快速排序

■算法描述

❖方法

$Divide: A[p..r] \Rightarrow A[p..q-1] \le A[q] \le A[q+1..r]$

$Conquer:$ 递归对$A[p..q-1]$，$A[q+1..r]$快排

终结条件，区间长度为1 时空操作

$Combine:$ 空操作

❖算法

```
QuickSort (A, p, r) {
   if (p < r){
      q ← Partition(A, p, r);  //划分元A[q]已正确
      QuickSort(A, p, q−1);
      QuickSort(A, q+1, r);
   }
}
```

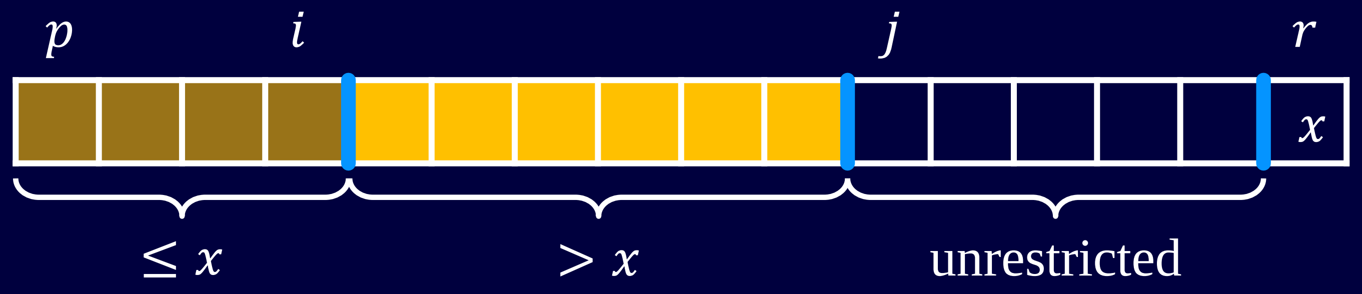
快速排序 (续)

❖划分算法

$$
\begin{aligned}
&Partition(A, p, r)\{\\
&\quad x \leftarrow A[r];\ \text{//区间最后元素为划分元}\\
&\quad i \leftarrow p-1;\\
&\quad for\ j \leftarrow p\ to\ r-1\ do\ \{\quad \text{//始终有}i \le j\\
&\quad\quad if\,(A[j] \le x)\{\quad \text{//使}A[p..i] \le x\\
&\quad\quad\quad i \leftarrow i+1;\\
&\quad\quad\quad A[i] \leftrightarrow A[j];\\
&\quad\quad\}\ \text{//end if 若含}A[j] > x\text{时，}j\text{加1}\\
&\quad\}\ \text{//end for}\\
&\quad A[i+1] \leftrightarrow A[r];\quad \text{//划分元位置为}i+1\\
&\quad return\ i+1;\\
&\}
\end{aligned}
$$

快速排序 (续)

❖ **循环不变量**



- $A[p..i] \leq x$
- $A[i+1..j-1] > x$
- $A[j..r-1]$ 尚未确定
- $A[r] = x$

- Note：

  这种划分使得：$A[p..q-1] \leq A[q] < A[q+1..r]$

快速排序 (续)

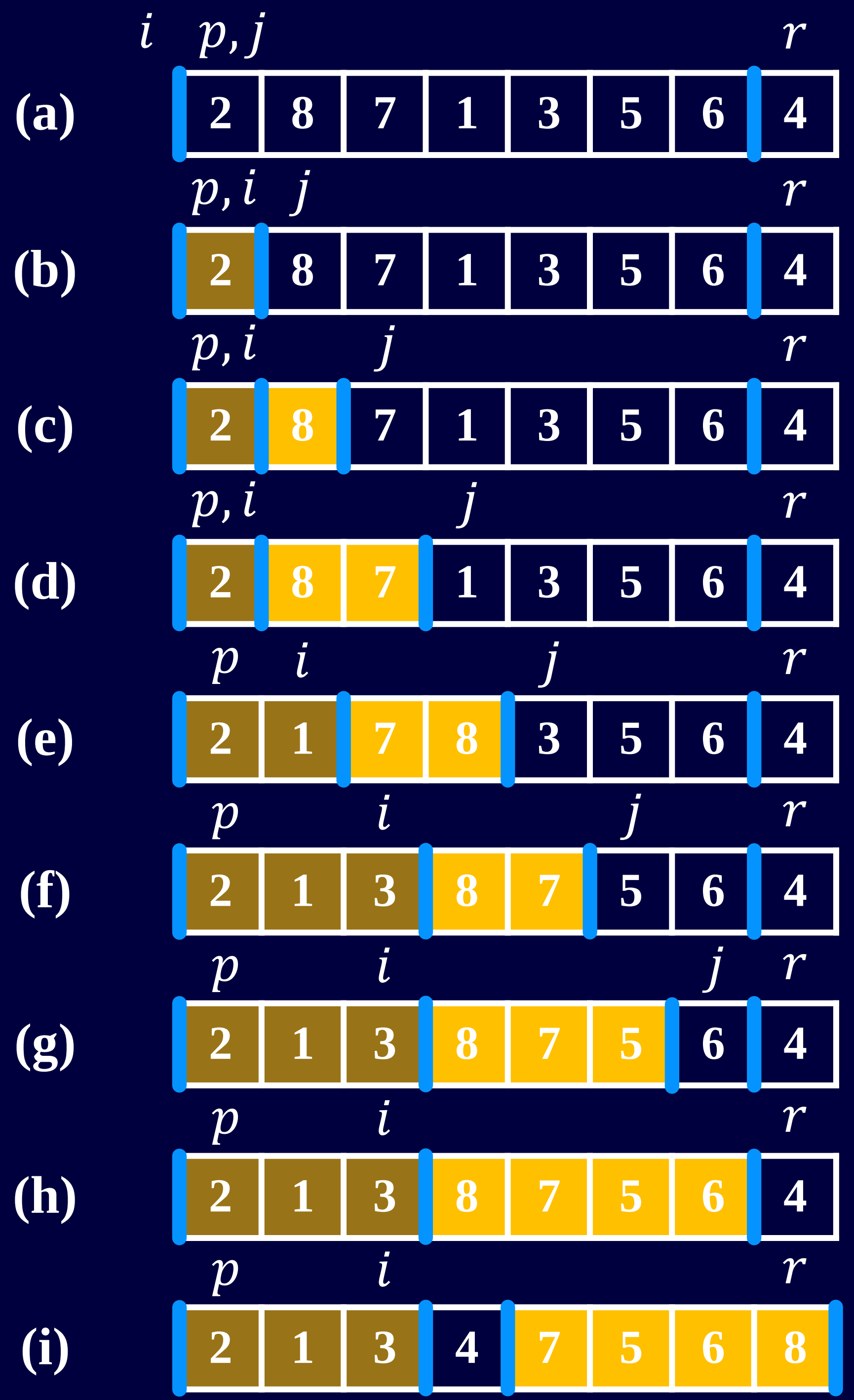
❖循环不变量

➢$A[p..i] \leq x$

➢$A[i+1..j-1] > x$

➢$A[j..r-1]$ 尚未确定

➢$A[r] = x$



快速排序 (续)

性能分析

划分是否平衡？

❖ **最坏划分：（已有序）**

$$T(n) = T(n-1) + T(0) + \Theta(n) \quad \text{//有一子区间为空}$$

$$= T(n-1) + \Theta(n)$$

$$= \sum_{k=1}^{n} \Theta(k) = \Theta(\sum_{k=1}^{n} k) = \Theta(n^2)$$

❖ **最好划分：（两子问题大小大致相等）**

$$T(n) = 2T(n/2) + \Theta(n)$$

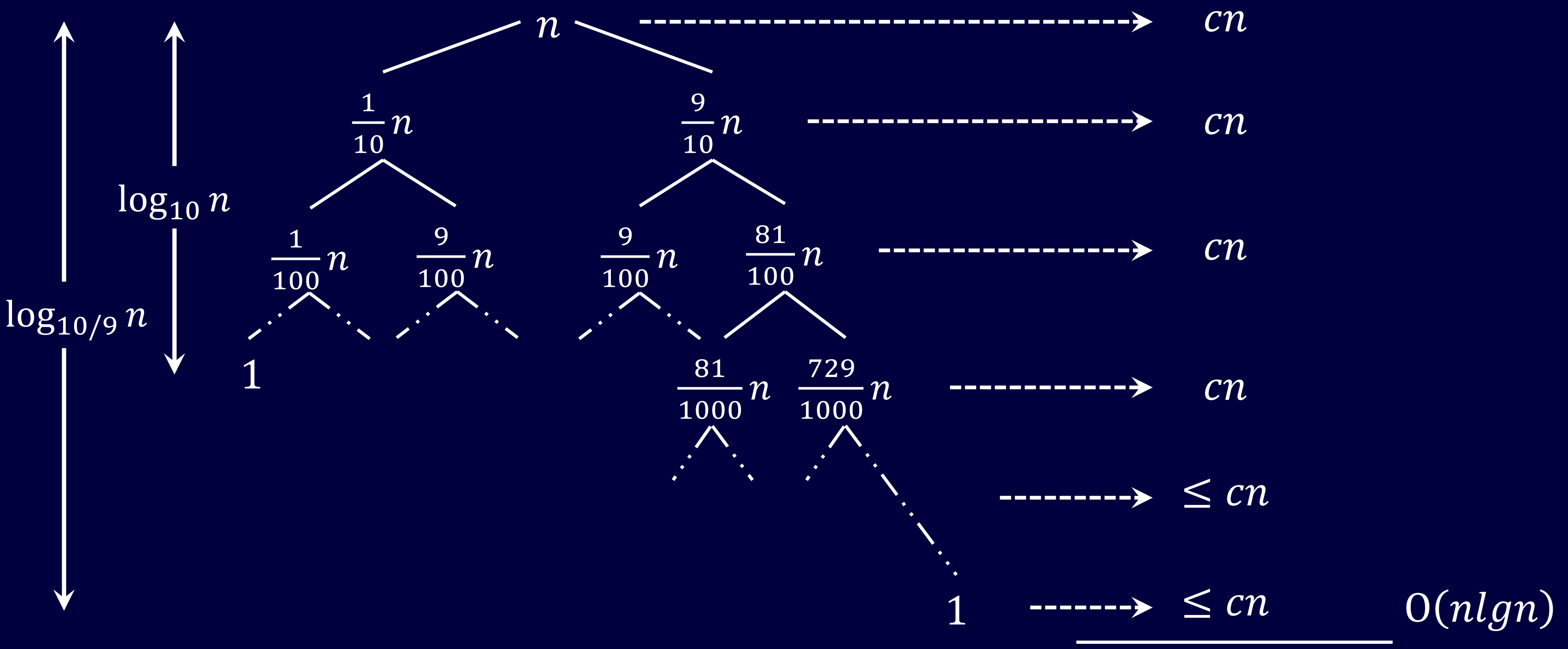
$$= \Theta(n \lg n) \quad \text{//由master定理}$$

快速排序 (续)

❖ 平衡划分

快排平均时间接近于最好情况，设划分总是产生9:1划分

$$T(n) \le T(9n/10) + T(n/10) + cn$$



快速排序 (续)

❖平衡划分 (续)

∵任何底大于1的对数与以1为底的对数之间只相差一常数因子

∴任何常数比例（如99:1, 999:1）的划分，树高仍为$O(lgn)$,

从而快排时间为$O(nlgn)$

尽管最坏时间是 $\Theta(n^2)$，但期望时间为 $\Theta(n\lg n)$

基于比较排序的时间下界为： $\lg n! = n\lg n - 1.44n + O(\lg n)$

快速排序平均：$1.39n\lg n + O(n)$，系数较小故称“快排”

快速排序 (续)

随机化版本

- 快速排序的平均性能假定：输入的所有排列是等可能的
- 算法随机化是指：算法行为不仅由输入确定，而且与随机数发生器产生的值有关。强迫输入分布是随机的

$RandomizedPartition\ (A, p, r)$ {//取代原partition

  $i \leftarrow Random(p, r);$ //在$[p..r]$中选随机数i

  $A[r] \leftrightarrow A[i];$

  $return\ \ Partition\ (A, p, r);$ //取$A[r]$作划分元

}

- 随机化算法分析较困难；但该算法非常有效，在排序过程中，某次随机选择最坏不会影响总体效果

快速排序 (续)

■ 期望运行时间

- ❖基本运算：划分子程序将划分元$A[r]$与$A[i]$进行比较
- ❖输入规模：n，数组A中的元素数
- ❖前提条件：没有理由认为数组中各元素的数值具有特定顺序，因此预设划分子程序返回的划分元位置取$1 \sim n$中任意值的概率相同

❖Note：

这一部分所分析的期望运行时间是对每种可能顺序的数组进行相同次数的排序时得到的平均排序时间

快速排序 (续)

■ 期望运行时间 (续)

❖ 引理：对于 $\begin{cases} t_n = t_{n-1} + {}^{2}/_{n} & (n > 1) \\ t_1 = 1 \end{cases}$，有 $t_n = 2\ln n$

❖ 求解过程：

$$\begin{aligned} t_n &= t_{n-1} + {}^{2}/_{n} \\ t_{n-1} &= t_{n-2} + {}^{2}/_{n-1} \\ t_{n-2} &= t_{n-3} + {}^{2}/_{n-2} \\ &\vdots \\ t_2 &= t_1 + {}^{2}/_{2} \\ t_1 &= 1 \end{aligned}$$

将后一个方程代入前一个方程

$$\begin{aligned} t_n &= t_{n-1} + {}^{2}/_{n} \\ &= t_{n-2} + {}^{2}/_{n-1} + {}^{2}/_{n} \\ &= t_{n-3} + {}^{2}/_{n-2} + {}^{2}/_{n-1} + {}^{2}/_{n} \\ &\vdots \\ &= t_1 + {}^{2}/_{2} + {}^{2}/_{3} + \cdots + {}^{2}/_{n-2} + {}^{2}/_{n-1} + {}^{2}/_{n} \\ &= 0 + {}^{2}/_{2} + {}^{2}/_{3} + \cdots + {}^{2}/_{n-2} + {}^{2}/_{n-1} + {}^{2}/_{n} \\ &= 2\sum_{i=2}^{n} {}^{1}/_{i} = 2\ln n \end{aligned}$$

快速排序 (续)

■ 期望运行时间 (续)

❖推导过程：

快排算法的平均时间复杂度可通过如下递推式计算

$$A(n) = \sum_{p=1}^{n} {}^{1}/_{n}\,[A(p-1) + A(n-p)] + (n-1) \quad (1)$$

易证，$\sum_{p=1}^{n}[A(p-1) + A(n-p)] = 2\sum_{p=1}^{n} A(p-1) \quad (*)$

将 $(*)$ 代入递推式有，

$$A(n) = {}^{2}/_{n} \sum_{p=1}^{n} A(p-1) + (n-1)$$

两边同时乘 n 有，

$$nA(n) = 2\sum_{p=1}^{n} A(p-1) + n(n-1) \quad (2)$$

快速排序 (续)

- 推导过程 (续):

为了得到 $A(n)$ 和 $A(n-1)$ 之间的关系，将 $n = \mathrm{n} - 1$ 代入方程 (2) 有，

$$(n-1)A(n-1) = 2\sum_{p=1}^{n-1} A(p-1) + (n-1)(n-2) \quad (3)$$

(2) − (3) 得，

$$nA(n) - (n-1)A(n-1) = 2A(n-1) + 2(n-1)$$

$$nA(n) - nA(n-1) = A(n-1) + 2(n-1)$$

化简得，

$$\frac{A(n)}{n+1} = \frac{A(n-1)}{n+1} + \frac{2(n-1)}{n(n+1)}$$

快速排序 (续)

- 推导过程 (续)：

根据上式，令 $a_n = \frac{A(n)}{n+1}$ 有，

$$a_n = a_{n-1} + \frac{2(n-1)}{n(n+1)}$$

根据引理可知，上递推式的通项公式为 $a_n \approx 2\ln n$。因此，

$$\begin{aligned} A(n) &= 2\ln n \cdot (n+1) \\ &= 2\ln 2 \cdot \lg n \cdot (n+1) \\ &\approx 1.38 \lg n \cdot (n+1) \in \Theta(nlgn) \end{aligned}$$